
Article

[Robert Cemper](#) · Mar 27, 2021 3m read

IRIS easy ECP workbench

Testing ECP-based applications often take quite some effort for setup and preparation.

I have created a Docker-based workbench that allows you to have it quick at hands.

And if you crash it? You just give your containers a fresh start.

The whole setup runs code-based during the start-up of your instance.

In that sense, it is also a portable coding example using ZPM and the [objectscript-docker-template](#)

[see Video](#)

It's a fast setup of ECP client / server with almost no manual intervention

with everything included in the repository available from OEX.

Except:

As the default community-license doesn't include multi-server-features (ECP) you might have to use your own license or get a temporary loan [license from WRC](#) that you add to the downloaded set from GitHub as ECP_{iris}.key before starting.

Installation:

From terminal/command prompt in the download directory run

docker-compose build

And it will prepare 2 containers for ECPserver and ECPclient.

They are pretty much the same except that they get a **hostname: server** and **client**

This allows communication between the containers based on network names

instead of dynamic (and unpredictable) DHCP addresses.

The internal ports for external access as SMP, ... 1972,52773,53773 are published to

41773,42773,43773 for the server and **41443,42443,43443** for the client

Next, you launch both containers by

docker-compose up -d server

docker-compose up -d client

Don't start containers together but in sequence, as it may take some time until the whole configuration completes [only during the first startup] of the container.

For a simple reason I had to learn:

The final hostname is not available in the temporary containers during the build!

And there is obviously a difference!

In addition, the client can only connect when the server finished startup.

I experienced up to 120 sec until ECP was running stable and both startups completed.

Impatient people (as me) meanwhile watch the progress by

docker logs <containername>

It is exactly the output you see in messages.log

Testing:

You may either open a session to server or client by

docker-compose exec server iris session iris

docker-compose exec client iris session iris

or you use SMP at local ports 42773 or 42443

Test data:

In namespace USER Code and Data are stored in separated databases:

Code (Classes) are in database USER in both containers.

Data are stored in the database IRISLOCALDATA on the server

On the client, data from the same database are used as RemoteDatabase over ECP

Classes and test data are borrowed from SAMPLES in Caché and preloaded.

- Sample.Address
- Sample.Company
- Sample.Customer
- Sample.Employee
- Sample.Person
- Sample.USZipCode
- Sample.Utills
- Sample.Vendor

So trying, testing, extending, screwing this workbench by your own Classes, Tables, Data, Queries are ready to go.

[GitHub](#)

[#Docker](#) [#ECP](#) [#Tips & Tricks](#) [#InterSystems IRIS](#)

Source URL: <https://community.intersystems.com/post/iris-easy-ecp-workbench>